Dynamic Visual Servoing with Image Moments for an Unmanned Aerial Vehicle Using a Virtual Spring Approach

Ryuta Ozawa, François Chaumette

Dynamic Visual Servoing with Image Moments for an Unmanned Aerial Vehicle Using a Virtual Spring Approach

Ryuta Ozawa[a], François Chaumette[b]

a *Department of Robotics, Ritsumeikan University, Shiga, 525-8577 JAPAN, and ryuta@se.ritsumei.ac.jp*

b *INRIA Rennes-Bretagne Atlantique and IRISA, 35 042 Rennes, FRANCE, and Francois.Chaumette@irisa.fr*

Abstract

This paper presents an image-based visual servoing for controlling the position and orientation of an unmanned aerial vehicle (UAV) using a fixed downward camera observing landmarks on the level ground. In the proposed method, the visual servoing of the image moments is used to control the vertical motion and rotation around the roll axis. In contrast, an undesired positive feedback arises in visual errors because of the under-actuation of the UAV and this positive feedback makes it difficult to apply the visual servoing to the horizontal motion. Thus, a novel control method using the virtual spring is introduced to control the horizontal motion. The stability of the system is proved based on Lyapunov's direct method. Simulations are presented to validate the proposed method.

keywords: Passivity-based control, image-based visual servoing, unmnanned aerial vehicles, under-actuated systems,

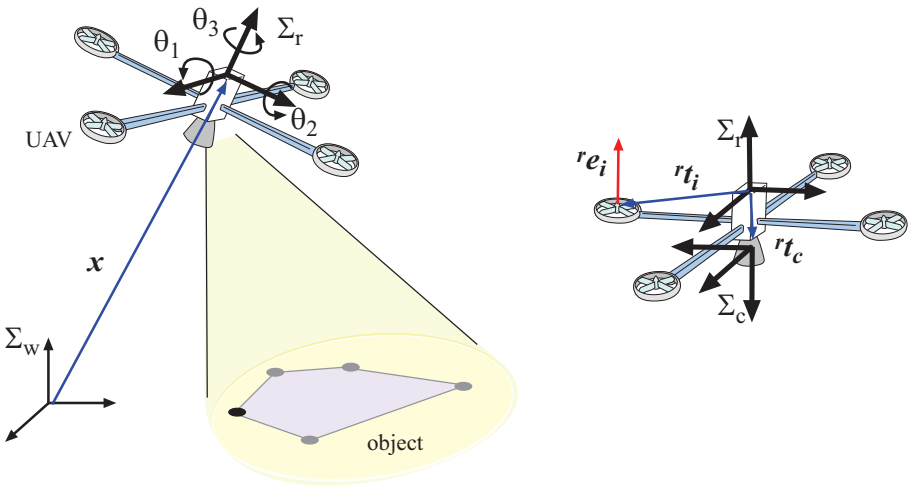

Fig. 1: Coordinates of a UAV.

1 Introduction

Demand of UAVs is growing last decade for civilian applications including search and rescue, wild fire monitoring, traffic monitoring, pipeline patrol and so on [1]. Attitude control [2] and pose control [3] of UAVs have become increasingly important to be applied to those applications. Gyroscopes are used for attitude control and inertial sensors or GPSs are often used to sense the current pose. However, it is difficult to obtain accurate current position and orientation using those sensors [4]. In contrast, a camera is useful in observing the current state and visual servoing is a powerful tool to control the pose using a camera. Visual servoing can be divided into two main classes [5]; position-based visual servoing (PBVS) and image-based visual servoing (IBVS). PBVS requires an accurate geometric model to estimate the current pose and is sensitive to image measurement errors and geometric model errors. On the other hand, IBVS uses image features directly, and is less sensitive to those errors than PBVS, though it has its own problems. It is generally more difficult to build controllers with IBVS than those with PBVS due to the complexity of kinematics between control variables and image features.

Much of the existing researches for visual servoing of UAVs have used PBVS in an eye-to-hand configuration. The position of a UAV was controlled using a feedback linearization and a backstepping method [6]. An image-based PID controller using a stationary camera has been proposed to control the pose of a helicopter under the assumption the roll and pitch motion can be neglected [7]. The position and orientation of multiple UAVs were dynamically controlled using the high speed tracking system [8]. A PBVS in an eye-in-hand configuration was executed to track a target that is manually selected during flight [9].

IBVS in an eye-in-hand camera configuration for under-actuated systems such as a quadrotor has been proposed using the backstepping method to guarantee the convergence of the position of the UAV [10], [11], [12].The key idea is a passivity-like properties of the spherical image and this properties helped to design the controllers using the backstepping method. These controllers require the translational velocity that is usually difficult to observe. Therefore, an optical flow was introduced to eliminate the sensing of the translational velocity [13] [14]. These existing IBVS approaches guarantee the convergence of the position of UAVs. However, the control of the orientation around the yaw axis required additional controllers or guaranteed the boundedness of the orientation. Bourquardez et al. [15] have compared several kinematic IBVS algorithms experimentally using a UAV. They found that an IBVS using perspective image moments realized the translational motions more stably than that using the spherical image moment did. In contrast, the stability problem in the case of the spherical image moment has been solved [10] but this problem in the case of the perspective image moments is still open.

This paper proposes an IBVS for controlling the position and the yaw rotation of a UAV to the desired ones. We assumed that the UAV has fixed propellers and a fixed camera directed downward to observe landmarks on the level ground, and perspective image moments can be computed from coplanar landmarks on the level ground [16]. The proposed method is designed based on the transpose Jacobian method [17]. However, the transpose Jacobian method cannot be directly applied to control

the translations in the horizontal plane due to the under-actuation of a UAV. Therefore, we employ two special methods to control the translations. One is to regard the translational forces, which are generated by the incline of the UAV, as a result of the visual servoing. However, these virtual translational forces increase the image errors [11] [18] and destabilize the UAV. Therefore, we introduce an original method called virtual spring approach that is based on the kinematic properties of a UAV to keep the incline of the UAV within a bounded region.

Some benefits of the proposed approach are given as follows: First, the proposed controller is designed based on the transpose Jacobian method that requires simple kinematics and the error sensing. The proposed controller only needs the camera and propeller models and the mass of the UAV as the parameters. Second, the proposed approach only observes the image, the angular velocity and the unit gravity vector. The unit gravity vector only requires the acceleration and gyro sensors to predict. In contrast, the standard IMU sensor system is used to sense the sensing of the complete orientation. The IMU sensor system is equipped with the magnetic field sensors in addition to these sensors for sensing the rotation around the yaw axis [19]. Therefore, the costs about the parameters and sensing in the proposed method are much less than other computational approaches [10].

Third, the proposed approach can control the yaw rotation of the UAV. This motion was assumed to be constant or remained within a bounded region [10], [12], [14].

Section 2 models a UAV with fixed propellers and Section 3 briefly explains the image moments and the interaction matrices [16]. Section 4 discusses the feature of the feedback and proposes the controller. Section 5 discusses the stability problem of the UAV. Simulation results are finally presented to validate the effectiveness and robustness of the controller in Section 6.

2 Modeling of a UAV

First, we consider the kinematics of a UAV as shown in Fig. 1. Let $\boldsymbol{x} = (x, y, z)$ and $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ be the position and the orientation of the UAV in a static reference frame Σ_w, where the notation $(a, b, \cdots, c)$ in lines expresses a column vector. Let ${}^w\boldsymbol{R}_r$ be the rotational matrix from Σ_w to Σ_r, referred to as roll, pitch, and yaw and given as follows:

$$ {}^w\boldsymbol{R}_r = \begin{bmatrix} c_{\theta_2}c_{\theta_3} & -c_{\theta_2}s_{\theta_3} & s_{\theta_2} \\ c_{\theta_1}s_{\theta_3} + s_{\theta_1}s_{\theta_2}c_{\theta_3} & c_{\theta_1}c_{\theta_3} - s_{\theta_1}s_{\theta_2}s_{\theta_3} & -s_{\theta_1}c_{\theta_2} \\ s_{\theta_1}s_{\theta_3} - c_{\theta_1}s_{\theta_2}c_{\theta_3} & s_{\theta_1}c_{\theta_3} + c_{\theta_1}s_{\theta_2}s_{\theta_3} & c_{\theta_1}c_{\theta_2} \end{bmatrix}, \tag{1} $$

where $s_{\theta_i} = \sin\theta_i$ and $c_{\theta_i} = \cos\theta_i$. The velocity of the UAV with respect to Σ_r can be described as ${}^r\boldsymbol{v} = {}^r\boldsymbol{R}_w(\boldsymbol{\theta})\dot{\boldsymbol{x}}$, and the angular velocity ${}^r\boldsymbol{\omega}$ of the UAV is given as

$$ [\,{}^r\boldsymbol{\omega}]_\times = {}^r\boldsymbol{R}_w\ {}^w\dot{\boldsymbol{R}}_r = -\,{}^r\dot{\boldsymbol{R}}_w\,{}^r\boldsymbol{R}_w{}^{\mathrm{T}}, \tag{2} $$

where $[\cdot]_\times$ is the skew-symmetric matrix that satisfies $[\boldsymbol{a}]_\times \boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$. The vector expression of the angular velocity (2) is given as

$$
\begin{aligned}
{}^r\boldsymbol{\omega} &= \boldsymbol{G}\dot{\boldsymbol{\theta}} \\
\text{with } \boldsymbol{G} &= \begin{bmatrix} 1 & 0 & s_{\theta_2} \\ 0 & c_{\theta_1} & -s_{\theta_1} c_{\theta_2} \\ 0 & s_{\theta_1} & c_{\theta_1} c_{\theta_2} \end{bmatrix}.
\end{aligned} \tag{3}
$$

Thus, the linear and angular velocity of the UAV is given as

$$
{}^r\dot{\boldsymbol{z}} = \begin{bmatrix} {}^r\boldsymbol{v} \\ {}^r\boldsymbol{\omega} \end{bmatrix} = \boldsymbol{V}\dot{\boldsymbol{z}}, \tag{4}
$$

where

$$
\boldsymbol{V} = \begin{bmatrix} {}^r\boldsymbol{R}_w & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G} \end{bmatrix}, \tag{5}
$$

and $\dot{\boldsymbol{z}} = (\dot{\boldsymbol{x}}, \dot{\boldsymbol{\theta}})$.

Next, we consider the position of each propeller $\boldsymbol{z}_i (i = 1, 2, \cdots, N)$ in Σ_w given as follows:

$$
\boldsymbol{z}_i = \boldsymbol{x} + {}^w\boldsymbol{R}_r \ {}^r\boldsymbol{t}_i \tag{6}
$$

where ${}^r\boldsymbol{t}_i$ is the position vector from the UAV center of mass to the center of propeller in Σ_r. Then, the velocity relationship between Σ_w and Σ_r is

$$
{}^r\dot{\boldsymbol{z}}_i = \begin{bmatrix} {}^r\boldsymbol{v}_i \\ {}^r\boldsymbol{\omega}_i \end{bmatrix} = \boldsymbol{S}_i \ {}^r\dot{\boldsymbol{z}}, \tag{7}
$$

where ${}^r\boldsymbol{v}_i$ and ${}^r\boldsymbol{\omega}_i$ are the translational and rotational velocities of ${}^r\boldsymbol{t}_i$ in Σ_r, and

$$
\boldsymbol{S}_i = \begin{bmatrix} \boldsymbol{I}_3 & -[\,{}^r\boldsymbol{t}_i\,]_\times \\ \boldsymbol{0} & \boldsymbol{I}_3 \end{bmatrix}. \tag{8}
$$

The force/torque ${}^r\boldsymbol{f}_i$ generated at each propeller in Σ_r is

$$
{}^r\boldsymbol{f}_i = \begin{bmatrix} {}^r\boldsymbol{e}_i \\ \kappa_i \ {}^r\boldsymbol{e}_i \end{bmatrix} f_i, \tag{9}
$$

where ${}^r\boldsymbol{e}_i$ is the direction of the axis of the propeller, f_i is the magnitude of force generated at the propeller and κ_i is the torque ratio to $\boldsymbol{f}_i$. This is the generalized formulation of the multiple fixed propeller force used in [3], [8], [20].

The Lagrangian of the UAV is given as follows:

$$
L = \frac{1}{2} m \| \ {}^r\boldsymbol{v} \|^2 + \frac{1}{2} \ {}^r\boldsymbol{\omega}^{\mathrm{T}} \hat{\boldsymbol{I}} \ {}^r\boldsymbol{\omega} + m\boldsymbol{g}^{\mathrm{T}}\boldsymbol{x} \tag{10}
$$

where m is the mass of the UAV, $\hat{\boldsymbol{I}}$ is the inertia moment, and $\boldsymbol{g} = (0, 0, -g_r)$ is the gravitational vector. By minimizing the variation

$$
\int_0^t \left\{ \delta L - \sum_{i=1}^{N} \delta \ {}^r\boldsymbol{z}_i^{\mathrm{T}} \ {}^r\boldsymbol{f}_i \right\} \mathrm{d}t = \int_0^t \left\{ \delta L - \sum_{i=1}^{N} \delta \boldsymbol{z}^{\mathrm{T}} \boldsymbol{V}^{\mathrm{T}} \boldsymbol{S}_i^{\mathrm{T}} \ {}^r\boldsymbol{f}_i \right\} \mathrm{d}t, \tag{11}
$$

and by adding damping terms, the dynamical equation of the UAV is given as follows [21]:

$$\begin{bmatrix} m\boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{H}\dot{\boldsymbol{\theta}} + \boldsymbol{S}\dot{\boldsymbol{\theta}} \end{bmatrix} - \begin{bmatrix} m\boldsymbol{g} \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{f}, \tag{12}$$

where $\boldsymbol{H} = \boldsymbol{G}^{\mathrm{T}}\hat{\boldsymbol{I}}\boldsymbol{G}$, $\boldsymbol{S}$ is the skew-symmetric matrix related to the Coriolis and centrifugal force, $\boldsymbol{f} = (f_1, f_2, \cdots, f_N)$ is the force generated by the propellers, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the translational and rotational friction components, which are modeled as the first and the second powers of the velocity. $\boldsymbol{A}$ is the transmission matrix defined by

$$\boldsymbol{A}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \end{bmatrix}, \text{where } \boldsymbol{a}_i = \begin{bmatrix} \boldsymbol{I}_3 \\ [\,{}^r\boldsymbol{t}_i\,]_\times + \kappa_i \boldsymbol{I}_3 \end{bmatrix} {}^r\boldsymbol{e}_i \ . \tag{13}$$

It is important to understand the feature of the driving force for designing a controller for the UAV. All the fixed propellers usually direct upward (i.e., ${}^r\boldsymbol{e}_i = {}^r\hat{\boldsymbol{e}}$, where ${}^r\hat{\boldsymbol{e}} = (0, 0, 1)$). Therefore, the translational force $\tau_1 = \sum_{i=1}^{N} f_i$ can be generated only in the z direction. The horizontal translation can be controlled by inclining the UAV. In contrast, if ${}^r\boldsymbol{t}_i$ and κ_i are selected appropriately, then any torque $\boldsymbol{\tau}_2$ can be generated. Therefore, the four directions except the horizontal translation can be controlled if the rank of $\boldsymbol{A}$ is four, which is the minimal number of the propellers of the UAV.

Therefore, Eq. (12) can be rewritten as follows:

$$\begin{bmatrix} m\boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{H}\dot{\boldsymbol{\theta}} + \boldsymbol{S}\dot{\boldsymbol{\theta}} \end{bmatrix} - \begin{bmatrix} m\boldsymbol{g} \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{u}, \tag{14}$$

where $\boldsymbol{u}$ is the four-dimensional driving force vector and is expressed as follows:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix} \boldsymbol{\tau} = \begin{bmatrix} {}^r\hat{\boldsymbol{e}} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 1} & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \boldsymbol{\tau}_2 \end{bmatrix}. \tag{15}$$

Note that the dynamical equations (14) is effective if $\|\theta_2\| < \pi/2$ (rad) because $\boldsymbol{G}$ in Eq. (3), which is included in $\boldsymbol{V}$, is singular at $\pm\pi/2$ (rad). The quaternion representation [2] is useful to avoid this singularity. However, the UAV lies in the vertical plane when $\theta_2 = \pm\pi/2$ and cannot be controlled anymore. Thus, in this paper, we treat the behavior of the UAV only in the region where $\|\theta_2\| < \pi/2$.

3 Image moments and their interaction matrices

First we consider the general framework of image errors, and interaction matrices [5]. Let $\boldsymbol{s}_i$ and $\boldsymbol{s}_i^*$ be the i th component of the current and the desired image feature, $\Delta\boldsymbol{s}_i = \boldsymbol{s}_i - \boldsymbol{s}_i^*$ be the image error, ${}^c\dot{\boldsymbol{z}} = ({}^c\boldsymbol{v},\ {}^c\boldsymbol{\omega})$ be the camera velocity in the camera coordinates Σ_c. Then,

$$\Delta\dot{\boldsymbol{s}}_i = \boldsymbol{L}_{s_i}\ {}^c\dot{\boldsymbol{z}} = \boldsymbol{L}_{s_i}\ {}^c\boldsymbol{S}_r\ {}^r\dot{\boldsymbol{z}} = \boldsymbol{P}_{s_i}\ {}^r\dot{\boldsymbol{z}}\,, \tag{16}$$

where the interaction matrix $\boldsymbol{L}_{si}$ and the transformation matrix ${}^c\boldsymbol{S}_r$ are given as follows:

$$\boldsymbol{L}_{si} = \begin{bmatrix} \boldsymbol{L}_{s_i 1} & \boldsymbol{L}_{s_i 2} \end{bmatrix}, \text{ and } {}^c\boldsymbol{S}_r = \begin{bmatrix} {}^c\boldsymbol{R}_r & -\,{}^c\boldsymbol{R}_r\,[\,{}^r\boldsymbol{t}_c\,]_\times \\ \boldsymbol{0} & {}^c\boldsymbol{R}_r \end{bmatrix}, \tag{17}$$

where ${}^c\boldsymbol{R}_r = \mathrm{diag.}(1,-1,-1)$ and ${}^r\boldsymbol{t}_c$ is the position vector of the camera from the origin of Σ_r w.r.t. Σ_r. Therefore,

$$\boldsymbol{P}_{i1} = \boldsymbol{L}_{s_i 1}\,{}^c\boldsymbol{R}_r, \text{ and } \boldsymbol{P}_{i2} = -\boldsymbol{L}_{s_i 1}\,{}^c\boldsymbol{R}_r[\,{}^r\boldsymbol{t}_c]_\times + \boldsymbol{L}_{s_i 2}\,{}^c\boldsymbol{R}_r\,.$$

Figure 2 shows the relationship of the projection. We can easily understand the Jacobian matrices and the velocities of the image, camera and the UAV expressed in the different coordinate systems.

We now give the interaction matrices of perspective image moments as derived in [16]. As will be described later, the translational parts of these matrices are independent each other, and this property is useful to design our new controller. Let n points form a landmark on the level ground. Then, the moments are defined by:

$$m_{ij} = \sum_{k=1}^{n} x_k^i y_k^j, \tag{18}$$

where (x_i, y_i) is the position of the i th landmark in image space. Let (x_g, y_g) be the center of gravity of the landmarks in the image. The centered moments are defined by:

$$\mu_{ij} = \sum_{k=1}^{n} (x_k - x_g)^i (y_k - y_g)^j, \tag{19}$$

where $x_g = m_{10}/n$ and $y_g = m_{01}/n$, $m_{00} = n$. Let a be defined by

$$a = \mu_{20} + \mu_{02}. \tag{20}$$

As visual features to control the translation, we choose x_n, y_n and a_n which are given by [16]

$$x_n = a_n x_g, \qquad y_n = a_n y_g, \qquad a_n = Z^* \sqrt{\frac{a^*}{a}}, \tag{21}$$

where the superscript $*$ expresses the quantity when the UAV is in the desired configuration. Then, we define a first image error as follows:

$$\Delta \boldsymbol{s}_h = \begin{bmatrix} \Delta x_n \\ \Delta y_n \end{bmatrix} = \begin{bmatrix} x_n - x_n^* \\ y_n - y_n^* \end{bmatrix}. \tag{22}$$

The interaction matrix of the image error for the horizontal motions is given as follows:

$$\Delta \dot{\boldsymbol{s}}_h = \boldsymbol{P}_{s_h}\,{}^r\dot{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{P}_{s_{h1}} & \boldsymbol{P}_{s_{h2}} \end{bmatrix}\,{}^r\dot{\boldsymbol{z}}\,. \tag{23}$$

When the points are coplanar and parallel to the image plane, $\boldsymbol{P}_{s_{hi}} (i = 1, 2)$ can be described as follows [16]:

$$\boldsymbol{P}_{s_{h1}} = -\boldsymbol{L}_{s_{h1}}\,{}^c\boldsymbol{R}_r, \text{ and } \boldsymbol{P}_{s_{h2}} = \boldsymbol{P}_{s_{h1}}[\,{}^r\boldsymbol{t}_c]_\times + \boldsymbol{L}_{s_{h2}}\,{}^c\boldsymbol{R}_r, \tag{24}$$

$$\text{with } \boldsymbol{L}_{s_{h1}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \text{ and } \boldsymbol{L}_{s_{h2}} = \begin{bmatrix} a_n\epsilon_{11} & -a_n(1+\epsilon_{12}) & y_n \\ a_n(1+\epsilon_{21}) & -a_n\epsilon_{22} & -x_n \end{bmatrix}, \tag{25}$$

where

$$\begin{cases} \epsilon_{11} & = n_{11} + x_g(y_g - \epsilon_{31}), \qquad \epsilon_{12} = n_{20} + x_g(x_g - \epsilon_{32}), \\ \epsilon_{21} & = n_{02} + y_g(y_g - \epsilon_{31}), \qquad \epsilon_{22} = n_{11} + y_g(x_g - \epsilon_{32}), \\ n_{ij} & = \mu_{ij}/m_{00}. \end{cases}$$

Due to the particular form of $\boldsymbol{L}_{s_{h1}}$ and ${}^{c}\boldsymbol{R}_{r}$, we can note that

$$\boldsymbol{P}_{s_{h1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \text{ and } \boldsymbol{P}_{s_{h2}} = \begin{bmatrix} a_n\epsilon_{11} & 1 + a_n(1 + \epsilon_{12}) & -y_n \\ 1 + a_n(1 + \epsilon_{21}) & a_n\epsilon_{22} & x_n \end{bmatrix}. \tag{26}$$

Next, we consider a second image error $\Delta\boldsymbol{s}_v$,

$$\Delta\boldsymbol{s}_v = \begin{bmatrix} \Delta a_n \\ \Delta\alpha \end{bmatrix} = \begin{bmatrix} a_n - a_n^* \\ \alpha - \alpha^* \end{bmatrix}, \tag{27}$$

where α is the object orientation angle defined as

$$\alpha = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right). \tag{28}$$

The interaction matrix of the image error $\Delta\boldsymbol{s}_v$ is given as follows:

$$\Delta\dot{\boldsymbol{s}}_v = \boldsymbol{P}_{s_v}\ {}^{r}\dot{\boldsymbol{z}}, = \begin{bmatrix} \boldsymbol{P}_{s_{v1}} & \boldsymbol{P}_{s_{v2}} \end{bmatrix}\ {}^{r}\dot{\boldsymbol{z}}. \tag{29}$$

When the points are coplanar and parallel to the image plane, $\boldsymbol{P}_{s_{vi}}$ is given as follows [16]:

$$\boldsymbol{P}_{s_{v1}} = -\boldsymbol{L}_{s_{v1}}\,{}^{c}\boldsymbol{R}_{r}, \text{ and } \boldsymbol{P}_{s_{v2}} = \boldsymbol{P}_{s_{v1}}[\,{}^{r}\boldsymbol{t}_c]_\times + \boldsymbol{L}_{s_{v2}}\,{}^{c}\boldsymbol{R}_{r}, \tag{30}$$

$$\text{with } \boldsymbol{L}_{s_{v1}} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \boldsymbol{L}_{s_{v2}} = \begin{bmatrix} -a_n\epsilon_{31} & a_n\epsilon_{32} & 0 \\ \alpha_{wx} & \alpha_{wy} & -1 \end{bmatrix}, \tag{31}$$

where

$$\epsilon_{31} = y_g + (y_g\mu_{02} + x_g\mu_{11} + \mu_{21} + \mu_{03})/a \qquad \text{and} \qquad \epsilon_{32} = x_g + (x_g\mu_{20} + y_g\mu_{11} + \mu_{12} + \mu_{30})/a,$$
$$\alpha_{wx} = (\beta[\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21})] + \gamma x_g[\mu_{02}(\mu_{20} - \mu_{02}) - 2\mu_{11}^2] + \gamma y_g\mu_{11}[\mu_{20} - \mu_{02}])/d,$$
$$\alpha_{wy} = (\beta[\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12})] + \gamma x_g\mu_{11}[\mu_{20} + \mu_{02}] + \gamma[\mu_{20}(\mu_{02} - \mu_{20}) - 2\mu_{11}^2])/d,$$
$$d = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2,$$

where $\beta = 4$ and $\gamma = 2$. Here again, we can note that the forms of $\boldsymbol{P}_{s_{v1}}$ and $\boldsymbol{P}_{s_{v2}}$ simplify to

$$\boldsymbol{P}_{s_{v1}} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \boldsymbol{P}_{s_{v2}} = \begin{bmatrix} -a_n\epsilon_{31} & -a_n\epsilon_{32} & 0 \\ \alpha_{wx} & -\alpha_{wy} & 1 \end{bmatrix}. \tag{32}$$

4 Controller Design for a UAV

The control objective is to design the control input $\boldsymbol{u}$ for stabilizing the state $\boldsymbol{p} = (x_n, y_n, a_n, \alpha, \theta_1, \theta_2)$ of the UAV to the desired point $\boldsymbol{p}^* = (x_n^*, y_n^*, a_n^*, \alpha^*, 0, 0)$. The desired visual features x_n^*, y_n^*, a_n^* and

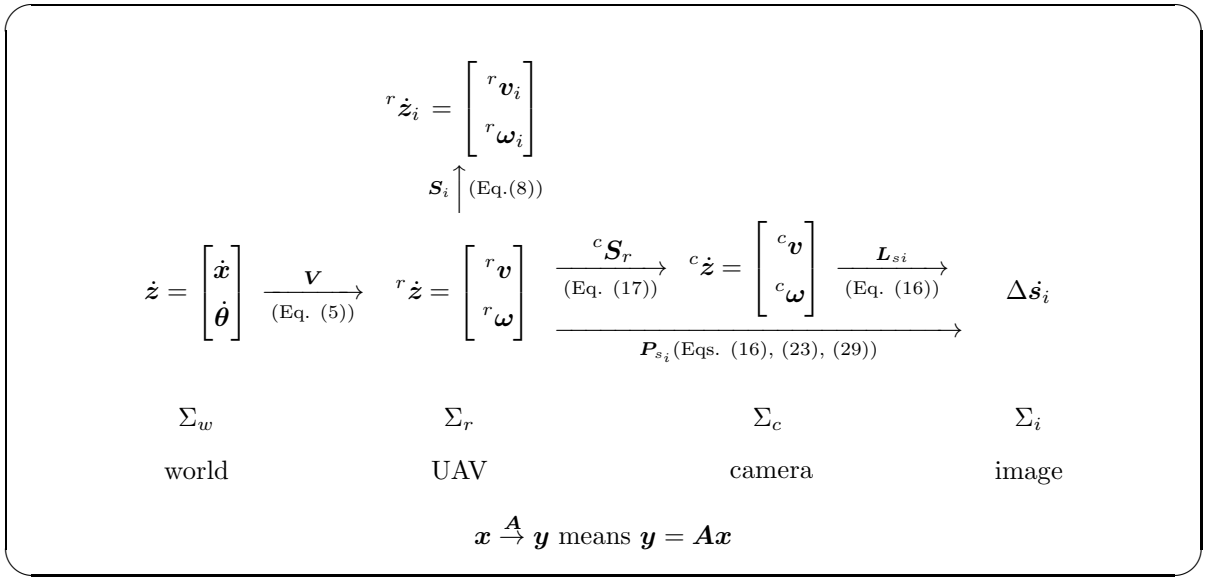

Fig. 2: Relationship of the velocities. $\dot{\boldsymbol{z}}$ and ${}^r\dot{\boldsymbol{z}}$ are the UAV velocities w.r.t. Σ_w and Σ_r, respectively. ${}^r\dot{\boldsymbol{z}}_i$ is the velocity of the origin of the propellers w.r.t. Σ_r. ${}^c\dot{\boldsymbol{z}}$ is the camera velocity w.r.t. Σ_c. $\Delta\boldsymbol{s}_i (i = h, v)$ is the velocity of the image error.

α^* can be chosen to reach any particular configuration of the UAV such that it is parallel to the level ground. We divide the controllers into the four parts as follows:

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{u}_1 + \boldsymbol{u}_2 + \boldsymbol{u}_3. \tag{33}$$

where $\boldsymbol{u}_0$ is the damping force and defined as

$$\boldsymbol{u}_0 = \begin{bmatrix} 0 \\ -\gamma\ {}^r\boldsymbol{\omega} \end{bmatrix}, \tag{34}$$

where γ is the feedback gain. ${}^r\boldsymbol{\omega}$ can be observed from the gyro sensor. $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are the visual servoing term in and around the z axis, and the term in the horizontal plane, respectively. $\boldsymbol{u}_3$ is a special spring term to stabilize the rotation around x and y axes. In the following subsections, we design each term separately and combine them to guarantee the convergence of the desired point.

4.1 Visual servoing in and around the z axis

We employ the transpose Jacobian method [17] for controlling the motion in and around the z axis using the visual error $\Delta\boldsymbol{s}_v$, as shown in Figure 3. Roughly speaking, the UAV approaches to the target if the observed area of the image is smaller than the desired one, and the UAV rotates around the z axis to eliminate the angle error calculated from the longest and shortest axes of the first-order image moments. The virtual potential V_1 composed of the square errors of the image moments in and around the z axis is given as follows:

$$V_1 = \frac{1}{2}\Delta\boldsymbol{s}_v^{\mathrm{T}}\boldsymbol{K}_v\Delta\boldsymbol{s}_v, \tag{35}$$

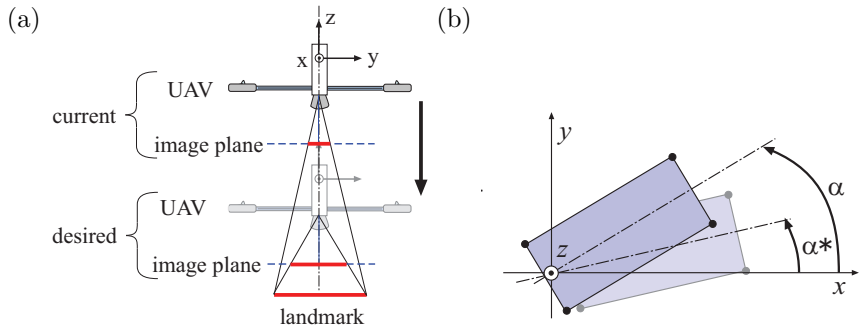

Fig. 3: Translational and rotational motions of the UAV in the vertical plane.(a) translational motion, (b) rotational motion.

where $\boldsymbol{K}_v = \mathrm{diag.}(k_{v1}, k_{v2})$ is a positive diagonal matrix. V_1 is regarded as a kind of potential functions, thus the control input to eliminate the image errors is given as follows:

$$\boldsymbol{u}_1 = -\frac{\partial V_1}{\partial\ {}^r\boldsymbol{z}}^{\mathrm{T}} = -\boldsymbol{P}_{s_v}^{\mathrm{T}}\boldsymbol{K}_v\Delta\boldsymbol{s}_v = \begin{bmatrix}\boldsymbol{u}_1^{\mathrm{tran}}\\ \boldsymbol{u}_1^{\mathrm{rot}}\end{bmatrix} = \begin{bmatrix}-\boldsymbol{P}_{s_{v1}}^{\mathrm{T}}\boldsymbol{K}_v\Delta\boldsymbol{s}_v\\ -\boldsymbol{P}_{s_{v2}}^{\mathrm{T}}\boldsymbol{K}_v\Delta\boldsymbol{s}_v\end{bmatrix} \tag{36}$$

The problem is that the translational motion of the UAV is under-actuated and we need to confirm whether $\boldsymbol{u}_1^{\mathrm{tran}}$ is in the range space of $\boldsymbol{B}_1$. The range space of $\boldsymbol{P}_{s_{v1}}^{\mathrm{T}}$ coincides with that of $\boldsymbol{B}_1$, because, from Eq. (32),

$$\boldsymbol{P}_{s_{v1}}^{\mathrm{T}} = \begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix} \tag{37}$$

and

$$\mathrm{span}\ \boldsymbol{B}_1 = \mathrm{span}\ \{\ {}^r\hat{\boldsymbol{e}}\} = \mathrm{span}\left\{\begin{bmatrix}0\\0\\1\end{bmatrix}\right\} \tag{38}$$

Therefore, $\boldsymbol{u}_1$ can be applied to control the translational and rotational z motions.

4.2 Visual servoing in the horizontal plane

Likewise in the previous subsection, we consider the following virtual potential V_2.

$$V_2 = \frac{1}{2}\Delta\boldsymbol{s}_h^{\mathrm{T}}\boldsymbol{K}_h\Delta\boldsymbol{s}_h, \tag{39}$$

where $\boldsymbol{K}_h$ is a positive diagonal matrix. Then, the control input for the motions in the horizontal plane is as follows:

$$-\frac{\partial V_2}{\partial\ {}^r\boldsymbol{z}}^{\mathrm{T}} = -\boldsymbol{P}_{s_h}^{\mathrm{T}}\boldsymbol{K}_h\Delta\boldsymbol{s}_h = \begin{bmatrix}-\boldsymbol{P}_{s_{h1}}^{\mathrm{T}}\\ -\boldsymbol{P}_{s_{h2}}^{\mathrm{T}}\end{bmatrix}\boldsymbol{K}_h\Delta\boldsymbol{s}_h. \tag{40}$$

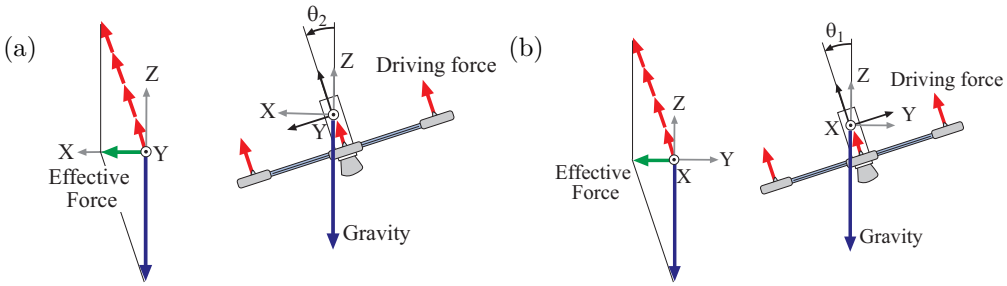

Fig. 4: The effect of Rotation for translational motion in horizontal plane. (a) x direction, (b) y direction

However, in this case, the translational part of Eq. (40) is not in the range space of ${}^{r}\hat{\boldsymbol{e}}$ because

$$\boldsymbol{P}_{s_{h1}}^{\mathrm{T}} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \notin \mathrm{span}\,\{\,{}^{r}\hat{\boldsymbol{e}}\} = \mathrm{span}\left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{41}$$

Therefore, we cannot realize this control input (40), and need to compensate this term using other effects. As shown in Fig. 4 (a), when the UAV inclines around the y axis and generates the gravity compensation force, the vertical force is canceled and the translational force in the x direction is generated implicitly. Then, the UAV can move in the x direction. (For y translational force, see Fig. 4 (b)). To realize this pure horizontal force, the propeller force in the z direction is given as follow:

$$\boldsymbol{u}_2^{\mathrm{tran}} = -\frac{mg_r}{{}^{r}\hat{\boldsymbol{e}}^{\mathrm{T}}\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}}\ {}^{r}\hat{\boldsymbol{e}}, \tag{42}$$

where ${}^{r}\boldsymbol{y}_{\mathrm{sen}}$ is the unit vector of the gravity, which is observed by the inertial sensors and is defined as follows:

$${}^{r}\boldsymbol{y}_{\mathrm{sen}} = -\,{}^{r}\boldsymbol{R}_{w}\ {}^{r}\hat{\boldsymbol{e}} = \begin{bmatrix} -s_{\theta_2} \\ s_{\theta_1} c_{\theta_2} \\ -c_{\theta_1} c_{\theta_2} \end{bmatrix}. \tag{43}$$

This term contains the roll and pitch angles of the UAV, while previous approaches need to sense ${}^{w}\boldsymbol{R}_{r}$, which requires an additional sensor such as a magnetic sensor to detect the yaw angle [6]-[12].

Then, the static horizontal force $\boldsymbol{h}_g$ is obtained by subtracting the gravity from Eq. (42) as follows:

$$\boldsymbol{h}_g = m\,{}^{r}\boldsymbol{R}_{w}\,\boldsymbol{g} - {}^{r}\hat{\boldsymbol{e}}\,\frac{mg_r}{{}^{r}\hat{\boldsymbol{e}}^{\mathrm{T}}\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}} = mg_r \left\{ {}^{r}\boldsymbol{y}_{\mathrm{sen}} - {}^{r}\hat{\boldsymbol{e}}\,\frac{1}{{}^{r}\hat{\boldsymbol{e}}^{\mathrm{T}}\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}} \right\}. \tag{44}$$

Here, we regard $\boldsymbol{h}_g$ as the resultant force of the first row of the visual feedback of Eq. (40). Then,

$$\boldsymbol{h}_g = -\boldsymbol{P}_{s_{h1}}^{\mathrm{T}} \boldsymbol{K}_h \Delta \boldsymbol{s}_h. \tag{45}$$

Therefore,

$$\boldsymbol{K}_h \Delta \boldsymbol{s}_h = -(\boldsymbol{P}_{s_{h1}}^{\mathrm{T}})^{+} \boldsymbol{h}_g, \tag{46}$$

where

$$(\boldsymbol{P}_{s_{h1}}^{\mathrm{T}})^{+} = (\boldsymbol{P}_{s_{h1}}\boldsymbol{P}_{s_{h1}}^{\mathrm{T}})^{-1}\boldsymbol{P}_{s_{h1}} = \boldsymbol{P}_{s_{h1}}. \tag{47}$$

The time derivative of $\boldsymbol{K}_h\Delta\boldsymbol{s}_h$ is given as follow:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{K}_h\Delta\boldsymbol{s}_h\right) = -(\boldsymbol{P}_{s_{h1}}^{\mathrm{T}})^{+}\frac{\partial \boldsymbol{h}_g}{\partial\ {}^{r}\boldsymbol{\omega}}\ {}^{r}\boldsymbol{\omega} = \boldsymbol{H}_h({}^{r}\boldsymbol{y}_{\mathrm{sen}})\ {}^{r}\boldsymbol{\omega}, \tag{48}$$

with

$$\boldsymbol{H}_h({}^{r}\boldsymbol{y}_{\mathrm{sen}}) = mg_r(\boldsymbol{P}_{s_{h1}}^{\mathrm{T}})^{+}\left\{\boldsymbol{I}_3 + \frac{1}{({}^{r}\hat{\boldsymbol{e}}^{\mathrm{T}}\ {}^{r}\boldsymbol{y}_{\mathrm{sen}})^2}\ {}^{r}\hat{\boldsymbol{e}}\ {}^{r}\hat{\boldsymbol{e}}^{\mathrm{T}}\right\}[\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}]_{\times}, \tag{49}$$

where, to derive $\boldsymbol{H}_h$, we used the time derivative of ${}^{r}\boldsymbol{y}_{[sen]}$ given by

$${}^{r}\dot{\boldsymbol{y}}_{\mathrm{sen}} = -\ {}^{r}\dot{\boldsymbol{R}}_w\ {}^{w}\boldsymbol{R}_r\ {}^{r}\boldsymbol{R}_w\ {}^{r}\hat{\boldsymbol{e}} = {}^{r}\boldsymbol{\omega}\times\ {}^{r}\boldsymbol{y}_{\mathrm{sen}} = -\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}\times\ {}^{r}\boldsymbol{\omega} = -[\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}]_{\times}\boldsymbol{G}\dot{\boldsymbol{\theta}} \tag{50}$$

Then, the feedback term of the image errors in the horizontal plane using the transpose Jacobian method is given as follows:

$$\boldsymbol{u}_2^{\mathrm{rot}} = -\boldsymbol{H}_h^{\mathrm{T}}\Delta\boldsymbol{s}_h. \tag{51}$$

Therefore,

$$\boldsymbol{u}_2 = \begin{bmatrix}\boldsymbol{u}_2^{\mathrm{tran}}\\ \boldsymbol{u}_2^{\mathrm{rot}}\end{bmatrix} = \begin{bmatrix}-\frac{mg_r}{{}^{r}\hat{\boldsymbol{e}}^{\mathrm{T}}\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}}\ {}^{r}\hat{\boldsymbol{e}}\\ -\boldsymbol{H}_h^{\mathrm{T}}\Delta\boldsymbol{s}_h\end{bmatrix}. \tag{52}$$

Note that we introduced $\boldsymbol{K}_h$ to define the virtual potential but never used this in the controller. Instead of that, we use the gravitational potential, which is equivalent to this virtual potential from Eq. (46).

To control the translational and rotational motions in and around the z axis in Fig. 3, we consider the case where the UAV is parallel to the level ground above the target. We will see in the simulation results that the controller is also effective when it is not the case.

4.3 A virtual spring approach

We now consider the relationship between the image error and the translational motion. As shown in Fig. 5, the UAV would like to move to the left for eliminating the image error (Phase 1). The UAV inclines to generate the positive force in x direction, and the image error becomes larger (Phase 2). Then, the UAV moves till the image error is eliminated (Phase 3). Thus, to generate the translational force in the x direction, the image error must be fed back positively. However, the positive visual feedback makes the system unstable, and we must prevent the UAV from giving the overthrow so that a virtual spring is added as shown in Fig. 6 (b) right. To realize the virtual spring effect, we consider the following artificial potential.

$$V_3 = \frac{1}{2}\Delta\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}{}^{\mathrm{T}}\boldsymbol{K}\Delta\ {}^{r}\boldsymbol{y}_{\mathrm{sen}}\ , \tag{53}$$

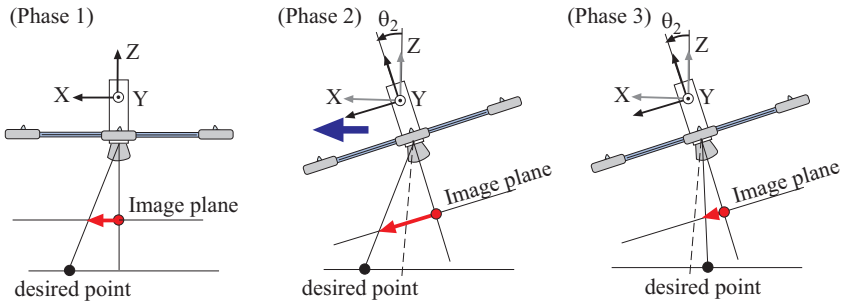

Fig. 5: Effect of Image errors for translational motion in horizontal plane. Positive feedback of the image errors inclines the UAV and generates the translational force in the horizontal direction. As a result, the image errors are eliminated by moving on the left.

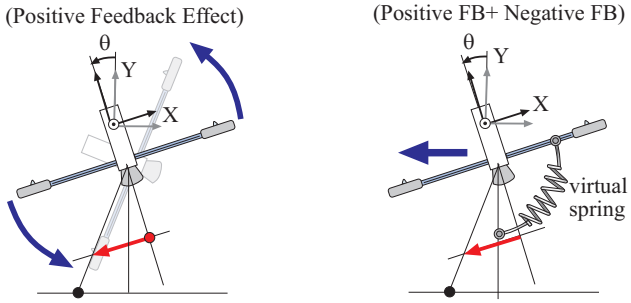

Fig. 6: Instability of the positive image feedback and the stabilization using the virtual spring. Positive visual feedback effect provides translational force, but the UAV over-rotates and the system become unstable. A virtual spring approach prevents from turning the UAV over, and is effective to eliminate the position errors.

where

$$\Delta\ {}^{r}\boldsymbol{y}_{\rm sen} = {}^{r}\boldsymbol{y}_{\rm sen} - \boldsymbol{e}_g. \tag{54}$$

Then, from Eq. (50),

$$\boldsymbol{u}_3 = -\frac{\partial V_3}{\partial\ {}^{r}\boldsymbol{z}}^{\rm T} = \begin{bmatrix} \boldsymbol{0}_{3\times 1} \\ [\ {}^{r}\boldsymbol{y}_{\rm sen}\ ]_\times \boldsymbol{K} \Delta\ {}^{r}\boldsymbol{y}_{\rm sen} \end{bmatrix}. \tag{55}$$

By substituting Eqs. (36), (52) and (55) into Eq. (33), the control input for the rotational part can be calculated as follows,

$$\boldsymbol{u} = \begin{bmatrix} -\boldsymbol{P}_{s_{v1}}^{\rm T} \boldsymbol{K}_v \Delta \boldsymbol{s}_v - \frac{mg_r}{{}^{r}\hat{\boldsymbol{e}}^{\rm T}\ {}^{r}\boldsymbol{y}_{\rm sen}}\ {}^{r}\hat{\boldsymbol{e}} \\ -\boldsymbol{P}_{s_v 2}^{\rm T} \boldsymbol{K}_v \Delta \boldsymbol{s}_v - \boldsymbol{H}_h^{\rm T}(\ {}^{r}\boldsymbol{y}_{\rm sen}\ ) \Delta \boldsymbol{s}_h + [\ {}^{r}\boldsymbol{y}_{\rm sen}\ ]_\times \boldsymbol{K} \Delta\ {}^{r}\boldsymbol{y}_{\rm sen} - \gamma\ {}^{r}\boldsymbol{\omega} \end{bmatrix}. \tag{56}$$

The damping term can be omitted when the UAV is enough damped by the natural damping $\boldsymbol{b}_i (i = 1, 2)$. Note that the controller (56) uses only the sensing of the landmarks, the gravity direction ${}^{r}\boldsymbol{y}_{\rm sen}$ and the angular velocity ${}^{r}\boldsymbol{\omega}$. In addition, the required parameters are only the mass of the UAV and the camera

internal parameters used in Eq. (56) and the propellers' information used in Eq. (13). Therefore, the implementation of this controller is easy.

The interaction matrices (23) and (29) are obtained under the assumption that the visual feature points lie in the plane parallel to the image plane. The robustness of these approximations will be confirmed in the simulations.

5 Stability Analysis

In this section, we prove the stability of the system (12) at the equilibrium point $(\boldsymbol{p}, \dot{\boldsymbol{z}}) = (\boldsymbol{p}^*, \boldsymbol{0})$ using the controller (56). Using Eq. (14) and (56), the closed loop of the system becomes

$$\begin{aligned} m\ddot{\boldsymbol{x}} &= -\boldsymbol{b}_1 + {}^w\boldsymbol{R}_r \left\{ \boldsymbol{h}_g - \boldsymbol{P}_{s_{v_1}}^{\mathrm{T}} \boldsymbol{K}_v \Delta \boldsymbol{s}_v \right\} \\ \boldsymbol{H}\ddot{\boldsymbol{\theta}} + \frac{1}{2}\dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}} &= -\boldsymbol{S}\dot{\boldsymbol{\theta}} - \boldsymbol{b}_2 - \boldsymbol{G}^{\mathrm{T}} \left\{ \boldsymbol{H}_h^{\mathrm{T}} \Delta \boldsymbol{s}_h + \boldsymbol{P}_{s_{v_2}}^{\mathrm{T}} \boldsymbol{K}_v \Delta \boldsymbol{s}_v + [\ {}^r\boldsymbol{y}_{\mathrm{sen}}\ ]_\times \boldsymbol{K} \Delta\ {}^r\boldsymbol{y}_{\mathrm{sen}} + \gamma\ {}^r\boldsymbol{\omega} \right\}. \end{aligned} \tag{57}$$

To prove the stability of the system, we choose the following as the candidate of the Lyapunov function,

$$V = \frac{1}{2} \left\{ m\|\dot{\boldsymbol{x}}\|^2 + \dot{\boldsymbol{\theta}}^{\mathrm{T}} \boldsymbol{H} \dot{\boldsymbol{\theta}} \right\} + \sum_{i=1}^{3} V_i, \tag{58}$$

where $V_i (i = 1, 2, 3)$ is defined in Eqs. (35), (39) and (53). This function is obviously positive definite. Then, the time derivative of V is given as follows:

$$\begin{aligned} \dot{V} =& m\dot{\boldsymbol{x}}^{\mathrm{T}}\ddot{\boldsymbol{x}} + \dot{\boldsymbol{\theta}}^{\mathrm{T}} \left(\boldsymbol{H}\ddot{\boldsymbol{\theta}} + \frac{1}{2}\dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}} \right) + {}^r\dot{\boldsymbol{v}}^{\mathrm{T}}\, \boldsymbol{P}_{s_{v1}}^{\mathrm{T}} \boldsymbol{K}_v \Delta \boldsymbol{s}_v + {}^r\boldsymbol{\omega}^{\mathrm{T}}\, \boldsymbol{P}_{s_{v2}}^{\mathrm{T}} \boldsymbol{K}_v \Delta \boldsymbol{s}_v + {}^r\boldsymbol{\omega}^{\mathrm{T}}\, \boldsymbol{H}_h^{\mathrm{T}} \Delta \boldsymbol{s}_h \\ & - {}^r\boldsymbol{\omega}^{\mathrm{T}}\, [\ {}^r\boldsymbol{y}_{\mathrm{sen}}\ ]_\times \boldsymbol{K} \Delta\ {}^r\boldsymbol{y}_{\mathrm{sen}} \\ =& -\boldsymbol{v}^{\mathrm{T}}\boldsymbol{b}_1 - \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{b}_2 - \gamma \parallel {}^r\boldsymbol{\omega} \parallel^2 \leq 0, \end{aligned} \tag{59}$$

where γ is positive scalar. The closed loop dynamics (57) was used in the derivation. The time derivative is semi-negative definite and the system is stable. From the LaSalle's invariant theorem [22], the state of the system that remains in a neighborhood of the desired point asymptotically approaches the set defined by

$${}^w\boldsymbol{R}_r \left\{ \boldsymbol{h}_g - \boldsymbol{P}_{s_{v_1}}^{\mathrm{T}} \boldsymbol{K}_v \Delta \boldsymbol{s}_v \right\} = \boldsymbol{0}, \tag{60}$$

$$\boldsymbol{G}^{\mathrm{T}} \left\{ \boldsymbol{H}_h^{\mathrm{T}} \Delta \boldsymbol{s}_h + \boldsymbol{P}_{s_{v_2}}^{\mathrm{T}} \boldsymbol{K}_v \Delta \boldsymbol{s}_v + [\ {}^r\boldsymbol{y}_{\mathrm{sen}}\ ]_\times \boldsymbol{K} \Delta\ {}^r\boldsymbol{y}_{\mathrm{sen}} \right\} = \boldsymbol{0}. \tag{61}$$

${}^w\boldsymbol{R}_r$ is non-singular, therefore, the vector in the parenthesis of Eq. (60) must be zeros. From (43) and (44), the vector in the parenthesis of Eq. (60) is given as follows:

$$\boldsymbol{h}_g - \boldsymbol{P}_{s_{v_1}}^{\mathrm{T}} \boldsymbol{K}_v \Delta \boldsymbol{s}_v = m g_r \begin{bmatrix} -\tan\theta_2 / \cos\theta_2 \\ \tan\theta_1 \\ 0 \end{bmatrix} - k_{v1} \Delta a_n \begin{bmatrix} s_{\theta_2} \\ -s_{\theta_1} c_{\theta_2} \\ c_{\theta_1} c_{\theta_2} \end{bmatrix} = \boldsymbol{0}. \tag{62}$$

$\Delta a_n = 0$ or $\theta_i = \pi/2 (i = 1, 2)$ from the last raw of (62). If $\theta_i = \pi/2$, then Δa_n becomes infinity from the first two rows. Δa_n and θ_i must be bounded, and this is contradicted. Thus, $\Delta a_n = 0$, and $\theta_1 = \theta_2 = 0$.

When $\theta_1 = \theta_2 = 0$, the matrix $\boldsymbol{G}$ is non-singular. Then, the vector in the parenthesis of Eq. (61) must be zeros. Substituting $\Delta a_n = \theta_1 = \theta_2 = 0$ int Eq. (61) and multiplying $(\boldsymbol{G}^{\mathrm{T}})^{-1}$ from the left, we obtain the following equation:

$$\boldsymbol{H}_h^{\mathrm{T}} \Delta \boldsymbol{s}_h + \boldsymbol{P}_{s_{h2}}^{\mathrm{T}} \boldsymbol{K}_v \begin{bmatrix} 0 \\ \Delta\alpha \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \Delta x_n \\ \Delta y_n \\ \Delta\alpha \end{bmatrix} = 0, \tag{63}$$

where

$$\boldsymbol{T} = \begin{bmatrix} 0 & mg_r & k_{v2}\alpha_{w1} \\ mg_r & 0 & -k_{v2}\alpha_{w2} \\ 0 & 0 & k_{v2} \end{bmatrix}. \tag{64}$$

x_n, y_n and α converge to zero because T is non-singular. Thus, we proved that the equilibrium point of the UAV is asymptotic stable.

6 Simulation

In this section, six selected simulations are presented to validate the effectiveness of the proposed controller using the conditions given in Table 1, and these gains were tuned through some preliminary simulations. The UAV has four propellers and its transmission matrix $\boldsymbol{A}$ is given as follows:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & -\ell & \kappa \\ 0 & 0 & 1 & \ell & 0 & -\kappa \\ 0 & 0 & 1 & 0 & \ell & \kappa \\ 0 & 0 & 1 & -\ell & 0 & -\kappa \end{bmatrix}, \tag{65}$$

where ℓ is the length of the moment arm, and κ is a ratio between the translational force to the rotational torque. The following parameters are used for the simulations; $m = 0.90$ (kg), $\ell = 0.30$ (m), $\hat{I} = (1.20, 1.20, 2.00) \times 10^{-3}(\mathrm{kgm}^2)$, and $\kappa = 0.0158$ (m). The damping terms are modeled as the quadratic form of the velocities arising from the aerodynamics [20] and defined as follows:

$$\boldsymbol{b}_1 = d_1 \boldsymbol{Q}({}^s\boldsymbol{v})\ {}^s\boldsymbol{v}, \boldsymbol{b}_2 = d_2 \boldsymbol{Q}({}^s\boldsymbol{v})\ {}^s\boldsymbol{v}, \tag{66}$$

where $\boldsymbol{Q}(a)$ is a diagonal matrix and the i-th diagonal element is absolute value of the i-th element of $\boldsymbol{a}$. The damping coefficients d_1 and d_2 are modeled as constant, based on experimental results in [23], and given in Table 2. The visual data is updated every 33 (ms) except Case 4, while ${}^r\boldsymbol{y}_{\mathrm{sen}}$ and $\dot{\boldsymbol{\theta}}$ are measured and the control input is updated every 1 (ms). We use four feature points, which are distributed around the origin of the base frame. As shown in Table 2, in Cases 1 to 4, the initial configuration is assigned so that the UAV goes from the vicinity to the above of the base frame. In Case 5, the initial configuration is assigned so that the UAV goes away from the above of the base frame. The differences in each configuration are shown in the image sampling time, the feedback gain γ. In

Case 6, a Gaussian noise with mean zero and standard deviation 2 pixels, which correspond to about 7.5 (mm) in the horizontal plane, 0.5 (m) above on the target, is added to the image. Other conditions are the same in Case 1. In Case 7, the UAV goes to the target on the same horizontal plane to validate the pure translational motions attained by the virtual spring approach. Other conditions are the same in Case 6.

Figs. 7-11 show the translation and orientation errors of each case as well as the image features error and the image points trajectory. In Case 1, a good damping is used as shown in Table 2, and the errors converge to zero around 15(s). In Case 2, less damping in the translational directions is used as shown in Table 2. Then, the system is under-damped and the convergence is retarded compared to Case 1. In Case 3, the damping feedback in the rotational directions is not used. Then, residual errors remain and the motion becomes vibrationally as in a limit cycle. This is due to the sampling time of the visual data. This motion is improved using 10 (ms) image sampling time as shown in Case 4, where the less damping in the translational directions and no damping feedback in the rotational directions are used, as well as using larger rotational damping as shown in Case 2.

In Case 5, the convergence is a little bit slower than in Case 1 because of the difference of the desired configuration. We can get the better performance when the landmarks are just below the UAV at a desired configuration. Indeed, the interaction matrix plays role of the moment arm of the image error, and the desired configuration in Case 5 becomes more sensitive to the image error than in Cases 1 to 4.

In Case 6, image errors appeared due to the introduction of the image noise. These errors induced some vibrations in the position and orientation of the UAV. However, the controller attenuated the vibrations and the magnitudes are smaller than those in the image. The proposed scheme is thus robust with respect to image noise.

In Case 7, the UAV approached to the target as the exponential maps and the image errors are almost straight lines.

7 Conclusion

This paper proposed an image-based visual servoing for controlling the position and the orientation of a UAV with a fixed camera, which points the downward direction. The transpose Jacobian control from image moments is used to control the translational and the rotational motions in the vertical axis. On the other hand, the lateral motion cannot be generated with this method due to the under-actuation of UAVs. Positive feedback structure of the image errors to the lateral motion was revealed, and a virtual spring was introduced to stabilize the pose of the UAV. Simulations showed that this method is effective to control the pose of a UAV, even if the low sampling rate of the visual data and the image errors may cause some residual errors. We show also that it is better to put the landmarks right below the desired pose of the UAV.

Future works will be devoted to realize experiments on a real platform. The current controller depends on the perspective image moment, and the controller is also undergoing to be generalized to

Table 1: Feedback gains used in the simulator

parameter	symbol	value
feedback gain of $\Delta \boldsymbol{s}_v$	$\boldsymbol{K}_v$	$1.00\boldsymbol{I}_2$
spring coefficient	$\boldsymbol{K}$	$10.0\boldsymbol{I}_3$

Table 2: Controller parameters. Point A is that $\boldsymbol{x}$ =(0, 0, 0.5)(m) and θ_3 =0(rad), Point B is that $\boldsymbol{x}$ =(0.5, 0.5, 5.5)(m) and $\theta_3 = 0.2\pi$(rad), Point C is that $\boldsymbol{x}$ =(0.0, 0.5, 5.5) and $\theta_3 = 0$ (rad).

	initial Point	final Point	natural damping	damping gain	image sample time	image noise
Symbol (Unit)			d_1, d_2 (kg/m), $\left(\frac{\mathrm{kgm}^2}{(\mathrm{rad})^2}\right)$	γ $\left(\frac{\mathrm{kgm}^2}{\mathrm{rad}\cdot\mathrm{s}}\right)$	(ms)	
Case 1	B	A	(100, 0.5)	1.5	33	none
Case 2	B	A	(10.0, 0.5)	1.5	33	none
Case 3	B	A	(100, 0.5)	0.0	33	none
Case 4	B	A	(10.0, 0.5)	0.0	10	none
Case 5	A	B	(100, 0.5)	1.5	33	none
Case 6	B	A	(100, 0.5)	1.5	33	Gaussian noise (s.d. 2 pixels)
Case 7	A	C	(100, 0.5)	1.5	33	Gaussian noise (s.d. 2 pixels)

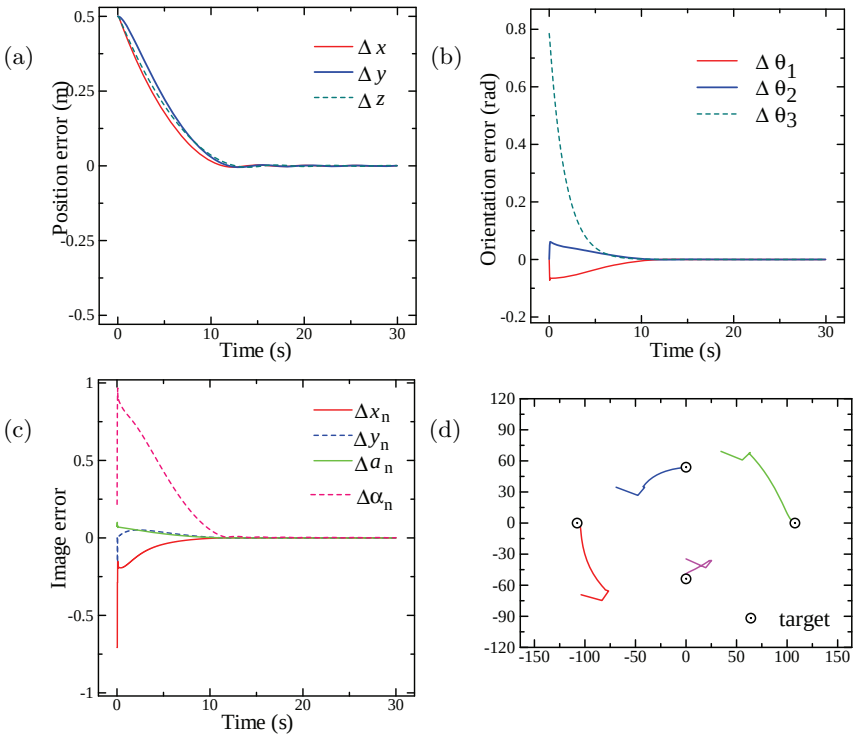

Fig. 7: Simulation results in Case 1. (a) Position error, (b) orientation error, (c) image error, and (d)image trajectories.

algorithms, which can use any other image features, and camera configuration.

Acknowledgment

We would like to thank Professor Tarek Hamel from I3S-CNRS, Nice-Sophia Antipolis, for his valuable comments and Dr. Nicolas Guenard from CEA/LIST for his valuable information of the parameters used in the simulation.

REFERENCES

[1] M. T. DeGarmo. Issues concerning integration of unmanned aerial vehicles in civil airspace. Technical report, MITRE, Center for Advanced Aviation System Development, McLean, Virginia, 2004.

[2] A. Tayebi and S. McGilvray. Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Trans. Control Syst. Technol.*, Vol. 14, No. 3, pp. 562–571, 2006.

[3] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic modelling and configuration stabilization for an X4-flyer. In *IFAC World Congress*, Barcelona, Spain, 2002.

[4] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics.* MIT press, 2006.

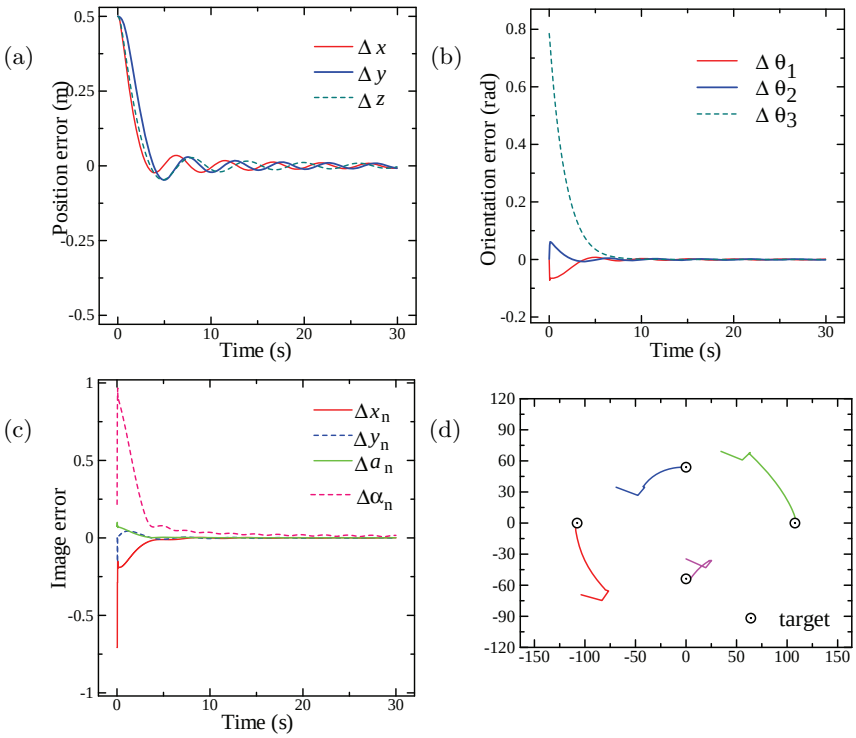

Fig. 8: Simulation results in Case 2, configured as Fig.7.

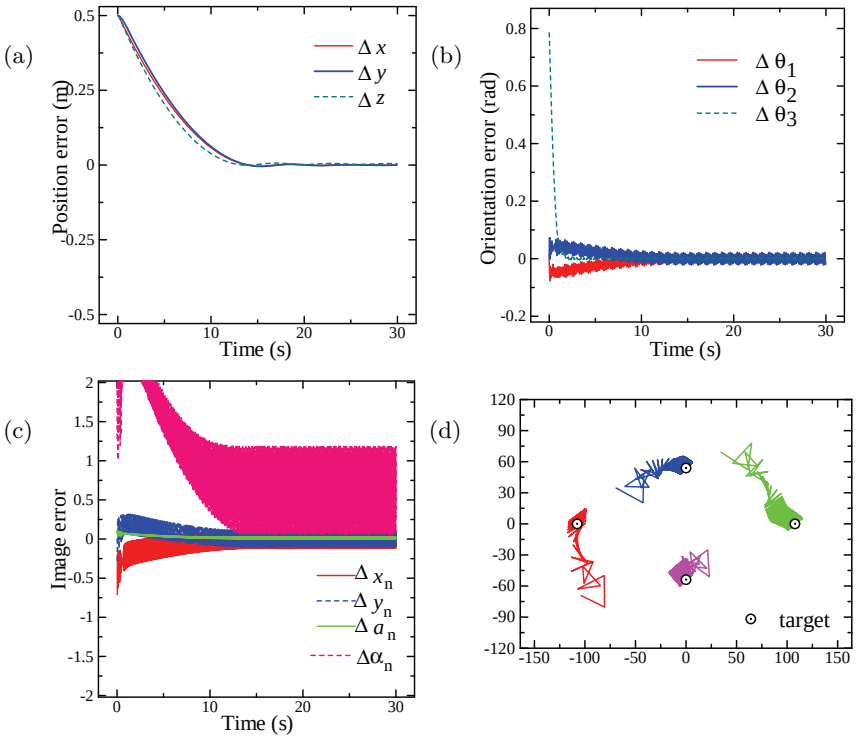

Fig. 9: Simulation results in Case 3, configured as Fig.7.

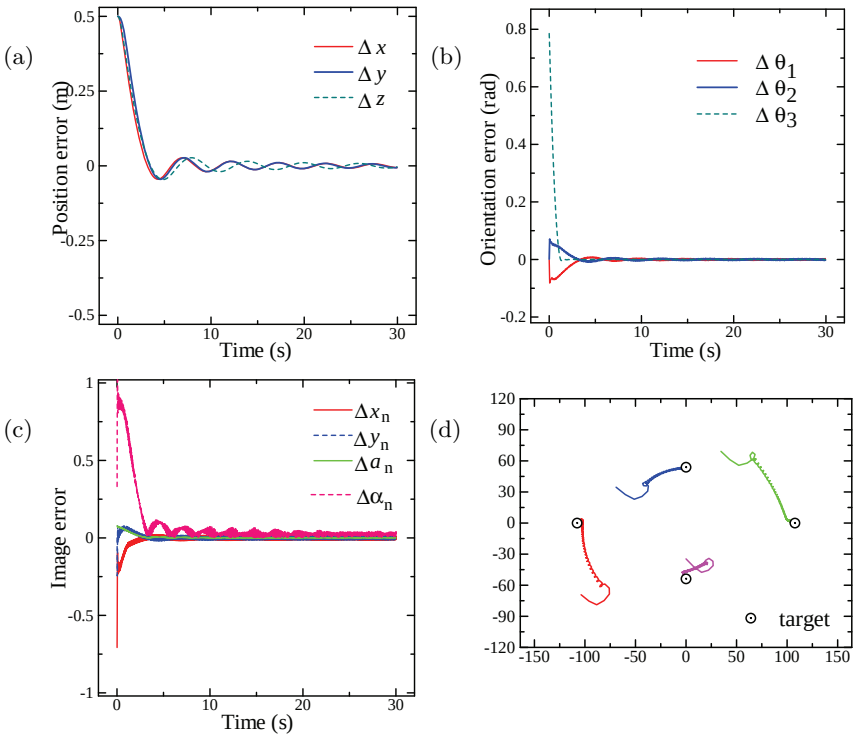

Fig. 10: Simulation results in Case 4, configured as Fig.7.

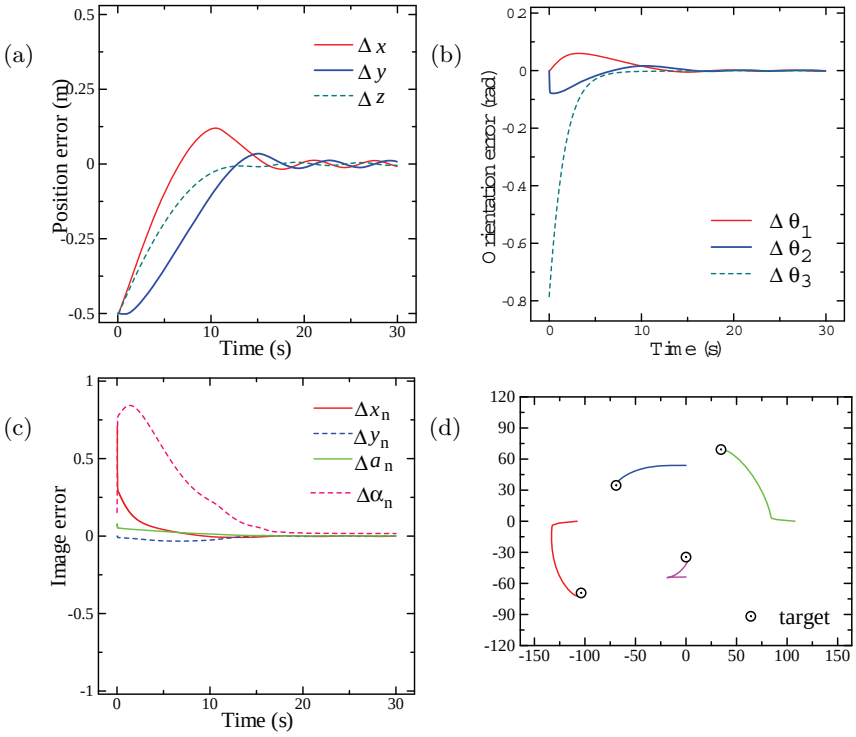

Fig. 11: Simulation results in Case 5, configured as Fig.7.

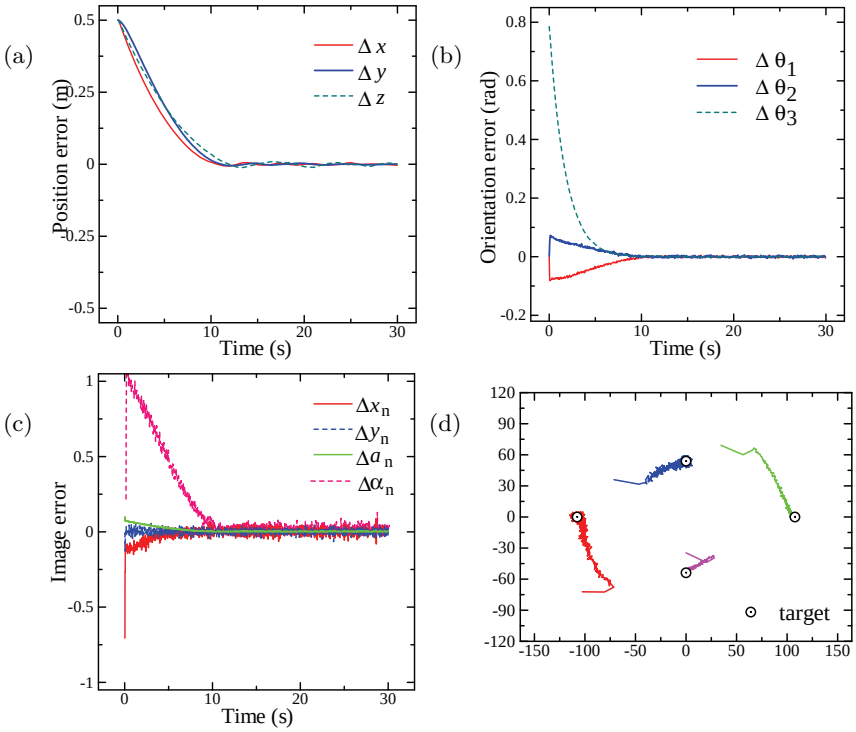

Fig. 12: Simulation results in Case 6, configured as Fig.7.

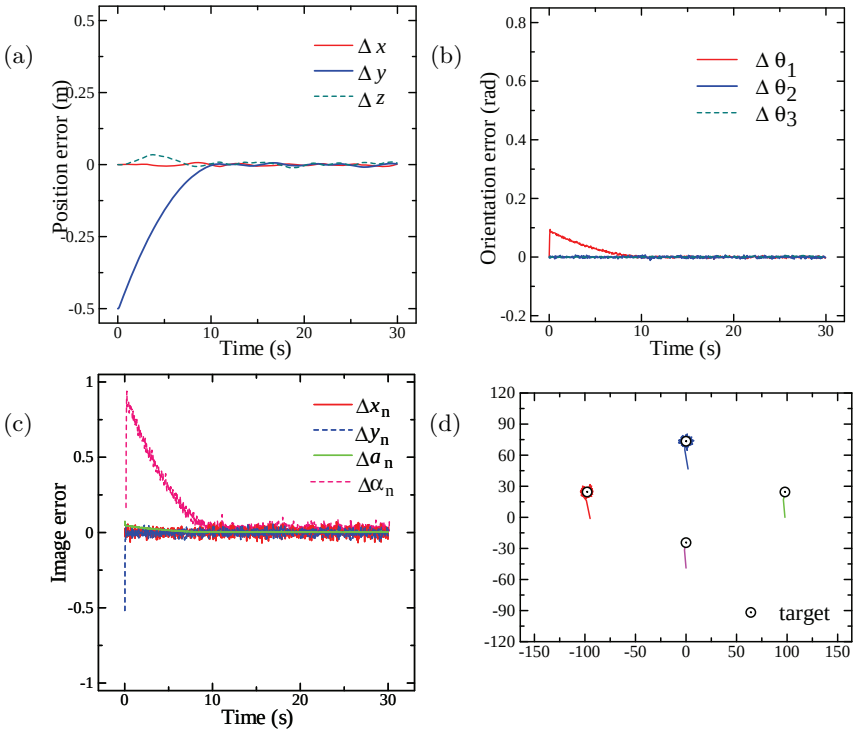

Fig. 13: Simulation results in Case 7, configured as Fig.7.

[5] F. Chaumette and S. Hutchinson. Visual servo control part I: Basic approaches. *IEEE Robot. Automat. Mag.*, Vol. 18, No. 4, pp. 82–90, December 2006.

[6] E. Altug, J. P. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *Proc. of IEEE Int. Conf. on Robot. and Autom.*, pp. 72–77, Washington, DC, May 2002.

[7] K. Watanabe, Y. Yoshihata, Y. Iwatani, and K. Hashimoto. Image-based visual PID control of a micro helicopter using a stationary camera. *Advanced Robotics*, Vol. 22, pp. 381–393, 2008.

[8] N. Michael, D. Mellinger, Q. Linsdsey, and V. Kumar. The GRASP multiple micro-UAV test bed. *IEEE Robot. Automat. Mag.*, Vol. 17, No. 3, pp. 56–65, 2010.

[9] S. Azrad, F. Kendoul, and K. Nonami. Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm. *J. of System, Design and Dynamics*, Vol. 4, No. 2, pp. 255–268, 2010.

[10] T. Hamel and R. Mahony. Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach. *IEEE Trans. on Robotics and Automation*, Vol. 18, No. 2, pp. 187–198, 2002.

[11] T. Hamel and R. Mahony. Image based visual servo control for a class of aerial robotic systems. *Automatica*, Vol. 43, pp. 1975–1983, 2007.

[12] N. Guenard, T. Hamel, and R. Mahony. A practical visual servo control for an unmanned aerial vehicle. *IEEE Trans. on Robotics*, Vol. 24, No. 2, pp. 331–340, 2008.

[13] R. Mahony, P. Corke, and T. Hamel. Dynamic image-based visual servo control usig centroid and optic flow features. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 130, pp. 011005.1–12, 2008.

[14] F. L. Bras, R. Mahony, T. Hamel, and P. Binetti. Dynamic image-based visual servo control for an aerial robot: theory and experiments. *International Journal of Optomechatronics*, Vol. 2, pp. 296–325, 2008.

[15] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck. Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Trans. on Robotics*, Vol. 25, , 2009.

[16] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. on Robotics*, Vol. 21, No. 6, pp. 1116–1127, 2005.

[17] M. Takegaki and S. Arimoto. A new feedback method for dynamic control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, pp. 119–125, 1981.

[18] R. Ozawa and F. Chaumette. Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach. In *Proc. of IEEE Int. Conf. on Robot. and Autom.*, pp. 5670–5676, Shanghai, CHINA, May 2011.

[19] X. Yun, E. R. Bachmann, and R. B. McGhee. A simplified quaternion-based algorithm for orientation estimation from earthgravity and magnetic field measurements. *IEEE. Trans. Instrum. Meas.*, Vol. 57, No. 3, pp. 638–650, 2008.

[20] R. Mahony, J. Kumar, and P. Corke. Multirotor aerial vehicles. *IEEE Robot. Automat. Mag.*, Vol. 19, No. 3, pp. 20–32, 2012.

[21] H. Goldstein, C. P. Poole, and J. L. Safko. *Classical Mechanics*. Pearson Education, 3rd edition, 2001.

[22] S. Arimoto. *Control theory of non-linear mechanical systems: a passivity-based and circuit-theoretic approach*. Oxford Science Publications. Oxford University Press, 1996.

[23] L. Derafa, T. Madani, and A. Benallegue. Dynamic modeling and experimental identification of four rotors helicopter parameters. In *Proc. of IEEE Inter. Conf. on Industrial Technology*, pp. 1834–1839, Dec. 2006.

About the Authors

Ryuta Ozawa received the B.S. degree in precision engineering and the M.S. and Ph.D. degrees in mechanical engineering from Meiji University, Kanagawa, Japan, in 1996, 1998, and 2001, respectively. From 1999 to 2002, he was a Research Assistant with Meiji University. In 2002, he was a Research Fellow with the Japan Society for the Promotion of Science. He is currently an Associate Professor with Ritsumeikan University, Kusatsu, Japan. His current research interests include tendon-driven mechanisms, elastic joint robots, visual servoing, and control of robotic hands and bipeds.

François Chaumette was graduated from École Nationale Supérieure de Mécanique, Nantes, France, in 1987. He received the Ph.D. degree in computer science from the University of Rennes, France, in 1990. Since 1990, he has been with Inria in Rennes where he is now "Directeur de Recherches" and head of the Lagadic group (`http://www.irisa.fr/lagadic`). His research interests include robotics and computer vision, especially visual servoing and active perception.

Dr. Chaumette received the AFCET/CNRS Prize for the best French thesis in automatic control in 1991. He also received with Ezio Malis the 2002 King-Sun Fu Memorial Best IEEE Transactions on Robotics and Automation Paper Award. He has been Associate Editor of the IEEE Transactions on Robotics from 2001 to 2005 and is now in the Editorial Board of the Int. Journal of Robotics Research.